

UML introduction I

Class Diagram & Sequence Diagram

利用UML的三个层面

- Conceptual View
- Specification View
- Implementation View

Conceptual View

特点:

1. 在非常high level 的层面---模块化层面
2. 很少或根本不考虑实现性
3. 完全与编程语言无关

Specification View

特点:

1. 在这个层面主要考虑软件的接口，类之间的接口 ---接口即是一种约定
2. 不关心实现，即对类的接口编程，而不要对类的实现编程

Implementation View

特点:

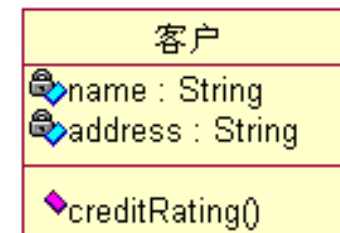
1. 类已定义好, 类之间关系也已定义好
2. 类 (UML图) 与实际代码间是精确相符的
3. 该层面的极致就是在各种tools的帮助下可以直接由相应的UML图生成code

Introduction

- Class Diagram
- Relationship between classes
- Sequence Diagram

Class 的表示

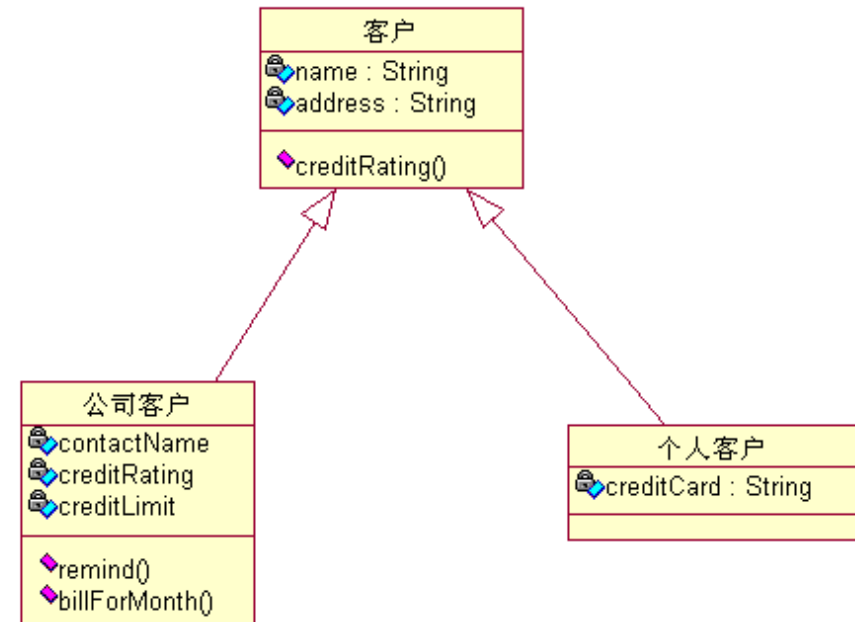
```
class 客户{  
    private:  
        string    name;  
        string    address;  
    public:  
        string    creditRating;  
};
```



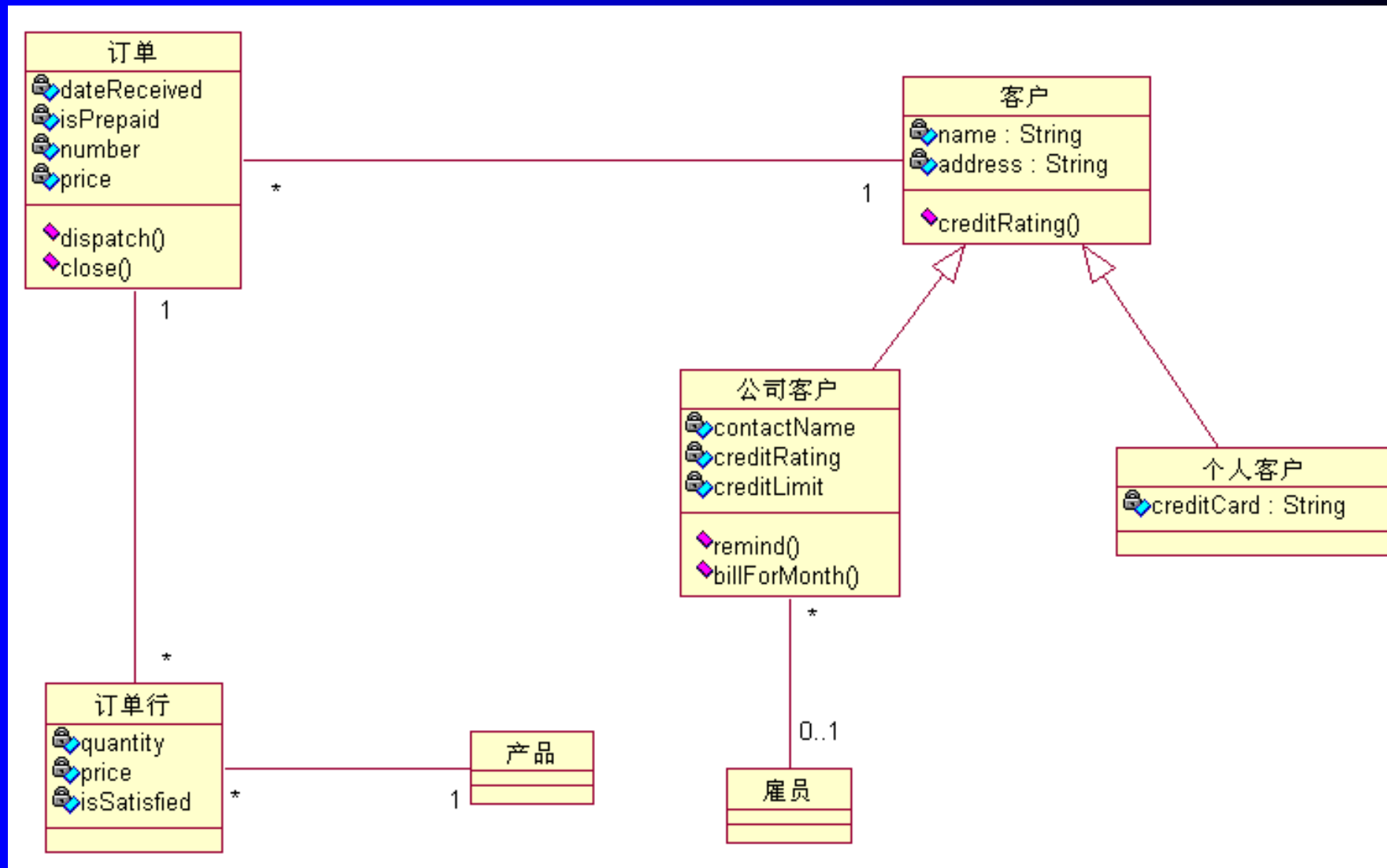
类继承的表示

```
class 个人客户: public客户{  
private:  
    string        creditCard;  
};
```

```
class 公司客户: public客户{  
private:  
    string        contactName;  
    string        creditRating;  
    string        creditLimit;  
public:  
    bool          remind();  
    bool          billForMonth();  
};
```



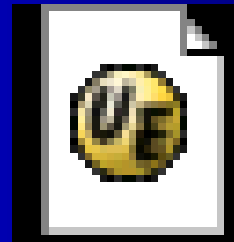
Relationship Between Classes



Relationship Between Classes

由伪代码来解释该图

请看UML-introduction-1.txt



UML-introduction-1.txt

类图及其关系在Rose中的操作

--- 在Rose中演示

UML中的sequence diagram

虚垂线的矩形小框---对象

垂线 ---对象的生命线

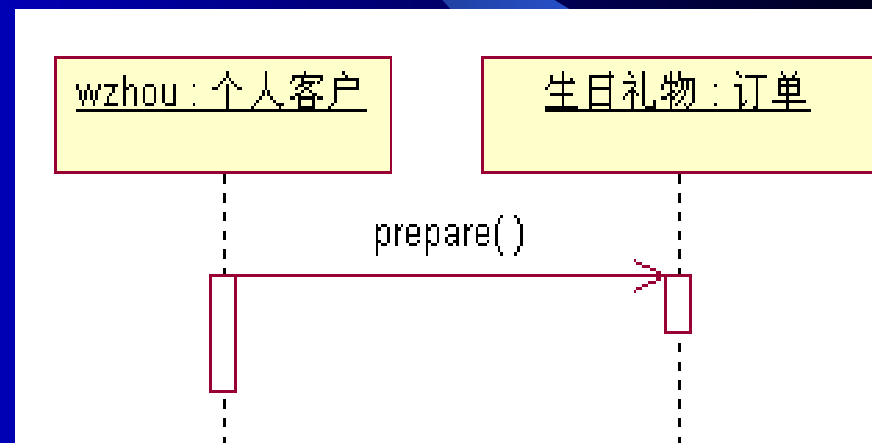
两对象生命线间的箭号--- 消息
(C++中的method)

对象名： 类名

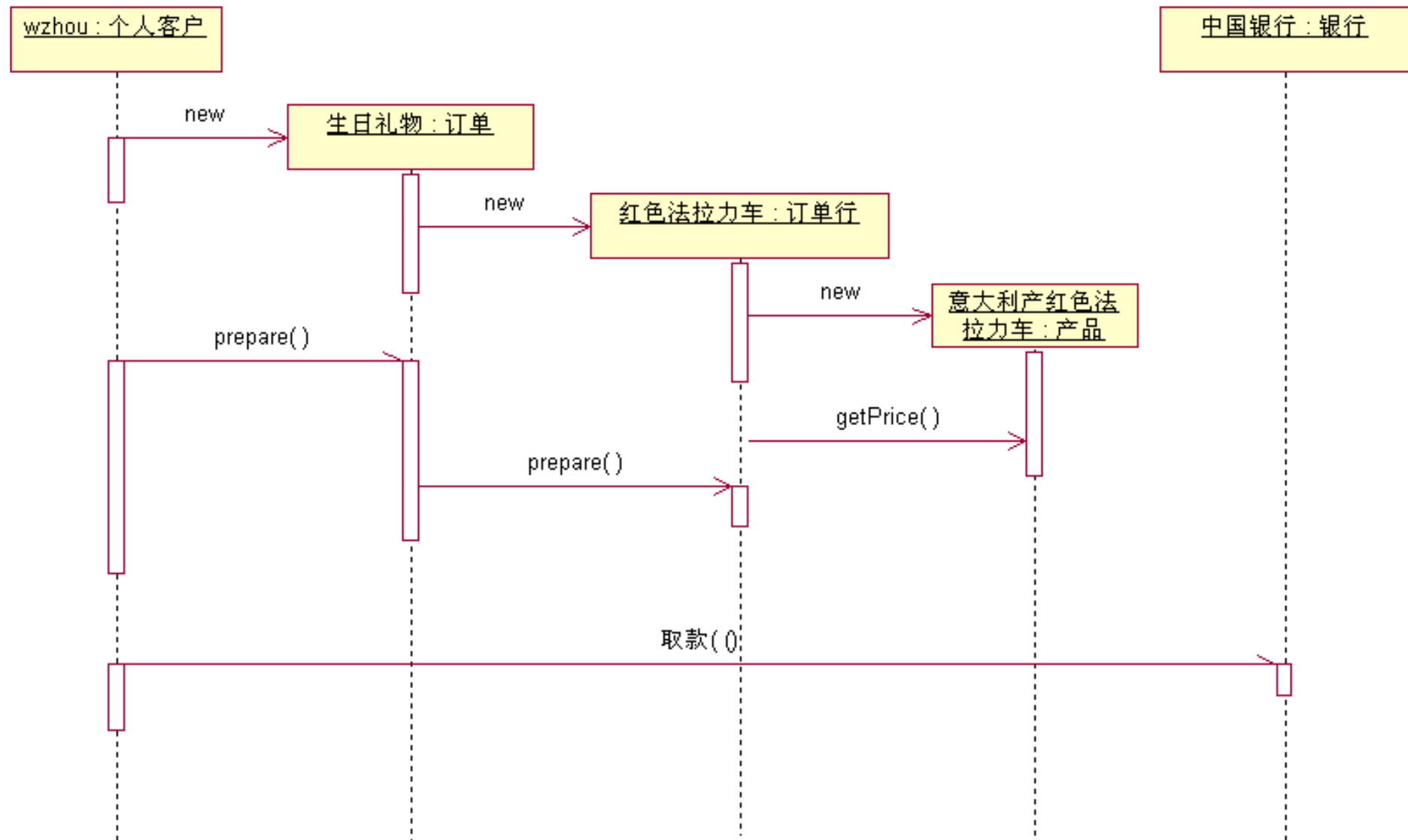
对象名 --- wzhou

类名 --- 个人客户

对象名可以省略，但：不能省略
(:个人客户)



Sequence Diagram Example



Sequence Diagram在Rose中的操作

--- 在Rose中演示